Fragebogen der Fachschaft zu **mündlichen Prüfungen** im Informatikstudium

Dieser Fragebogen gibt den Studierenden, die nach Dir die Prüfung ablegen wollen, einen Einblick in Ablauf und Inhalt der Prüfung. Das erleichtert die Vorbereitung.
Bitte verwende zum Ausfüllen einen schwarzen Stift. Das erleichtert das Einscannen.

Barcode:

Dein Studiengang: Informatik (MA)

Prüfungsart:
☐ Wahlpflichtfach
☒ Vertiefungsfach
☐ Ergänzungsfach

Welches? Probabilistische Planung

Prüfungsdatum: 04.08.2016

Prüfer/-in: Dr. Marco Huber

Beisitzer/-in: mir unbekannt

Prüfungsfächer und Vorbereitung:

Veranstaltung	Dozent/-in	Jahr	regelmäßig besucht?
Probabilistische Planung	Dr. Huber	SS 2016	Ja

Note: 1,0

Prüfungsdauer: 45 Minuten

War diese Note angemessen? Ja

✎ Wie war der **Prüfungsstil des Prüfers / der Prüferin?**
(Prüfungsatmosphäre, (un)klare Fragestellungen, Frage nach Einzelheiten oder eher größeren Zusammenhängen, kamen häufiger Zwischenfragen oder ließ er/sie dich erzählen, wurde Dir weitergeholfen, wurde in Wissenslücken gebohrt?)
Die Fragen waren klar gestellt. Die Atmosphäre war angenehm; er hat einen viel erzählen lassen. Ich konnte das meiste in Ruhe aufschreiben (einmal hat er gesagt, dass ich die Formel nicht aufschreiben muss) und er hat auch immer Feedback gegeben, dass ich das erzähle was er hören will. Die Fragen waren eigentlich immer klar; bei einer unklaren Frage habe ich direkt nachgehakt ob er XY meint und er hat es auch direkt bejaht. Super angenehm!

↪**Rückseite bitte nicht vergessen!**

✎ Hat sich der **Besuch / Nichtbesuch** der Veranstaltung für dich gelohnt?
Ja. Teilweise ist die Schrift schwer zu lesen, aber die Zusammenhänge werden klarer und Dr. Huber geht auch wunderbar auf Fragen ein.

✎ Kannst du ihn/sie **weiterempfehlen**? ☒ Ja / ☐ Nein
Warum?
Sehr nett, angenehme Athmosphäre.

✎ Wie lange und wie hast du dich **alleine bzw. mit anderen vorbereitet**?
Ich habe die Vorlesung 2 mal gehört, mich ca. 2 Monate immer wieder ein bisschen (ca. 2h/Tag) und ca. 2 Wochen intensiv (5h/Tag) vorbereitet. 3 Treffen à ca. 4h mit einem Lernpartner.

✎ Fanden vor der Prüfung **Absprachen** zu Form oder Inhalt statt? Wurden sie **eingehalten**?
Ja. Es wurde gesagt, dass keine Beweise dran kommen. War auch so.

✎ Welche **Tips zur Vorbereitung** kannst du geben? (Wichtige / Unwichtige Teile des Stoffes, gute Bücher / Skripten, Lernstil)
Folien lesen und verstehen, Protokolle durchgehen und meinen Blog lesen:
martin-thoma.com/probabilistische-planung/
Insbesondere die Tabelle am Ende, wo MDP / POMDP / RL verglichen werden sollte man auswendig können und aus dem FF beherrschen.

✎ Kannst du Ratschläge für das **Verhalten in der Prüfung** geben?
Mit den Antworten kann man etwas lenken, was als nächstes gefragt wird. Wenn man kurz Nachdenken muss, kann man das auch einfach sagen.

Inhalte der Prüfung:

Gedächtnisprotokoll; ich habe sicherlich ein paar Fragen / Details vergessen.

- Welche 3 Themen hatten wir in der Vorlesung

⇒ MDP (Markov Decision Processes), POMDP (Partially observable MDPs), RL (Reinforcement Learning). Ich habe auch gleich die Agent-Umwelt-Diagramme gezeichnet und daran die Unterschiede erklärt und habe das Explorationsproblem erwähnt.

- Gut. Zuvor hatten wir die Grundlagen mit Wahrscheinlichkeitstheorie, Optimierungs- und Nutzentheorie. Schreiben sie mir doch mal ein allgemeines Optimierungsproblem auf.

⇒

$$\arg\min_{x \in \mathbb{R}^n} f(x) \tag{1}$$
$$\text{s.t. } g_i(x) \leq 0 \quad \text{mit } i = 1, \ldots, m \tag{2}$$
$$h_j(x) = 0 \quad \text{mit } j = 1, \ldots, p \tag{3}$$

Siehe auch: https://martin-thoma.com/optimization-basics/.
Ich habe auch gleich erklärt warum $= 0$ genügt und warum man o.B.d.A. von einem Minimierungsproblem ausgehen kann.

- Ok, und was macht man wenn man Gleichungs-Nebenbedingungen hat?

⇒ Lagrange-Ansatz:

$$\mathcal{L}(x, \lambda_1, \ldots, \lambda_p) = f(x) + \sum_{j=1}^{p} \lambda_j \cdot h_j(x)$$

wobei das nun die notwendigen Nebenbedingungen für ein Optimum liefert, wenn man den Gradienten nach x und den Gradienten nach λ bildet und gleich 0 setzt.

- Was passiert bei den Gradienten nach λ?

⇒ Die Gleichungsnebenbedingungen kommen raus.

- Nun kam noch die Sache mit den Höhenlinien / den Gradienten und dem Winkel.

- Ok, verlassen wir die Optimierungstheorie. Was können sie zum Optimalitätsprinzip sagen?

⇒ Wenn man ein Problem mit optimaler Substruktur hat, dann gilt für jede optimale Lösung, dass die Lösungen der enthaltenen Teilprobleme optimal sein müssen. Sehr schön kann man das bei der kürzesten Wegesuche sehen.

- Zeigen sie das mal an einem Beispiel.

⇒ Wenn der kürzeste Weg von A nach E über B, C, D führt, dann muss der kürzeste Weg von B nach D auch über C führen. Falls das nicht so wäre — es also einen kürzesten Weg z.B. direkt von B nach D geben würde, dann wäre auch der Weg von A nach E kürzer wernn man direkt von B nach D gehen würde.

- Was hat das mit MDPs zu tun?

⇒ Anwendung findet es im Dynamic Programming (Endliche MDPs mit endlichem Horizont). Dabei geht man Rückwärtsrekursiv vor um die Wertefunktion J aus der Kostenfunktion g zu berechnen:

$$J(x_N) = g_N(x_N) \tag{4}$$
$$J(x_k) = \min_{a_k} \left[g_k(a_k, x_k) + \mathbb{E}\{J_{k+1}(x_k + 1) | x_k, a_k\} \right] \tag{5}$$

- Sehr schön, da haben wir auch gleich die Bellman-Gleichungen. Nun hatten wir noch geschlossen lösbare Spezialfälle. Welche sind das?

⇒ (i) Lineare Probleme (LQR) (ii) Endliche, deterministische Probleme (Label-Korrektur) (iii) Endliche Probleme mit unendlichem Horizont (Fixpunktsatz, Werteiteration, Bellman-Operator)

- Dann erklären Sie doch mal den LQR.

⇒ Zustandsraummodell ist linear und rauschen ist $r \sim \mathcal{N}(0, \Sigma)$:

$$x_{k+1} = A_k x_k + B_k a_k + r$$

Objective function ist:

$$\mathbb{E}\left(\underbrace{x_N^T \cdot Q_N \cdot x_N + \sum_{k=0}^{N-1} x_k^T \cdot Q_k \cdot x_k}_{\text{Zustandsabhängige Kosten}} + \underbrace{\sum_{k=0}^{N-1} a_k^T \cdot R_k \cdot a_k}_{\text{aktionsabhängige Kosten}}\right)$$

Der LQR ist dann einfach

$$a_k^* = \underbrace{-\left(R_k + B_k^T P_{k+1} B_k\right)^{-1} \cdot B_k^T \cdot P_{k+1} \cdot A_k}_{\text{Verstärkungsmatrix } L_k} x_k$$

wobei P_k Rückwärtsrekursiv durch die iterativen Riccati-Gleichungen bestimmt werden kann. (Hier wollte ich die aufschreiben, aber bei $P_N = Q_N$ hat er mich gestoppt.)

- Ok, das ist schon gut so. Nur Qualitativ, was machen die Riccati-Gleichungen?

⇒ Strukturell sind sie identisch zum Update der Fehlermatrix im Kalman-Filter duch den Update und Prädiktionsschritt.

- Ok, gut. Kommen wir zu POMDPs. Wie löst man die?

⇒ Belief-State MDP und Informationsvektor-MDP erklärt, Approximative Lösungen (Abbildung auf geschlossen lösbare Spezialfälle, Funktionsapproximatoren, Änderung der Optimierung)

- Ok. Warum verwendet man in der Praxis eher nicht das Informationsvektor-MDP?

⇒ Weil der Zustand in jedem Zeitschritt wächst. In jedem Zeitschritt k kommt eine weitere Aktion a_k hinzu; ggf. auch noch Beobachtungen z_k. Will man alles nutzen wird das Programm immer langsamer.

- Sie haben hinreichende Statistiken erwähnt. Was ist das?

⇒ (Definition; vgl. mein Blog-Artikel)

- Welche geschlossenen Spezialfälle gibt es bei POMDPs?

⇒ Linear (Kalman-Filter + LQR) und endlich (α-Vektoren)

- Was ändert sich beim LQR im POMDP-Fall?

⇒ $a_k = L_k \cdot \mathbb{E}(x)$

- Warum ist der Kalman-Filter toll?

⇒ Er erfüllt die BLUE-Eigenschaft (Best linear unbiased estimator). Das bedeutet, unter den erwartungstreuen linearen Schätzern ist er derjenige, welcher die geringste Varianz aufweist.

- Welche Annahmen machen wir beim Kalman-Filter?

⇒ Additives, mittelwertfreies normalverteiltes Rauschen und ein linearer Zustandsübergang.

- Was passiert, wenn das Rauschen nicht mehr normalverteilt ist?

⇒ Man muss die Kovarianz-Matrix berechnen können. Wenn das geht, dann ist der Kalman-filter immer noch der beste lineare Filter (aber es gibt nicht-lineare Filter die besser sind).

- Welche Bedingung muss der Zustandsschätzer für den LQR erfüllen?

⇒ Er muss erwartungstreu sein, was der Kalman-filter ja ist.

- Was bedeutet PWLC?

⇒ Piece-wise linear and concave. Da wir in der Vorlesung Minimierungsprobleme hatten, war es concave und nicht konvex. PWLC sind bei endlichen POMDPs die Wertefunktionen J_k (Zeichnung des Belief-State / der Aktionen; vgl. Links in meinem Blog). Ich habe noch Pruning erwähnt.

- Wie kann man einfach Pruning durchführen?

⇒ Es handelt sich um einen Simplex. (Beispiel mit nur 2 Zuständen aufgezeichnet.) Ein paarweiser Vergleich ist möglich, indem man nur die Endpunkte betrachtet. Wird eine Aktion echt von einer anderen dominiert, so kann diese Entfernt werden. Wird eine Aktion durch Kombinationen von Aktionen dominiert, so könnte man z.B. Algorithmen zur berechnun der Konvexen Hülle nutzen.

- Wie steigt die komplexität des α-Algorithmus in jedem Zeitschritt?

⇒ Exponentiell (in jedem Zeitschritt sind alle Aktionen prinzipiell wieder möglich)

- Ok, nun zu RL. Welche 3 Gruppen von Lösungsalgorithmen hatten wir?

⇒ Modellbasiert, Wertefunktionsbasiert, Strategiesuche. Modellbasiert kann mittels DP zu Wertefunktionsbasiert reduziert werden. Mit argmax kann man dann eine Strategie berechnen. Modellbasiert gibt es Dyna-Q, Adaptive DP und PILCO. Wertefunktionsbasiert hatten wir die Monte Carlo-Verfahren, Temporal Difference und die Kombination mit Eligibility traces.

- Was ist das Exploitation vs. Exploration-Problem?

⇒ Im RL kennen wir das Modell nicht. Wir befinden uns sozusagen im Dunkeln und müssen erst finden wo es Rewards gibt. Am Anfang muss man also Explorieren. (Habe eine Grid-World gezeichet und eine Pfad, wo ein Roboter einen Reward von 100 gefunden hat). Nun könnte man die Strategie so aufbauen, dass immer dieser Pfad (versucht wird) zu nehmen. Allerdings kann man auch darauf hoffen, dass an anderer Stelle (eingezeichnet) ein Reward von z.B. 150 ist. Um das herauszufinden muss man von der aktuell „optimalen“ Strategie abweichen und explorieren.

- Wie macht man das?

⇒ Durch probabilistische Strategien. Das einfachste ist, dass man am Anfang $\varepsilon \in \mathbb{N}$ Schritte exploriert und dann deterministisch die Strategie benutzt. Besser sind GLIE-Strategien, die theoretisch unendlich oft alle Zustände besuchen. Nennenswert sind ε-Greedy und Softmax.

- Zeichnen sie die Verteilung mal auf, wenn sie 3 Aktionen haben und Aktion 1 optimal ist, Aktion 2 die zweitbeste und Aktion 3 die schlechteste.

⇒ Es ergibt sich, dass bei ε-Greedy die nicht-optimalen Aktionen gleichverteilt sind und bei softmax ist die Verteilung vom aktuell geschätzten Wert der Aktion abhängig. Da gibt es noch eine Temperatur τ, welche mit der Zeit sinkt. Am Anfang ist der Q-Wert der Aktionen also nicht so wichtig, aber später mehr. Es gibt noch ausgefeiltere Explorations-Strategien welche berücksichtigen wie viel sich in der Q-Funktion noch ändert.

- Ok, dass hatten wir nicht in der Vorlesung. Damit ist die Zeit auch schon rum.